



A time-aware resource search strategy with the ant colony optimization in MANETs

Xianfu Meng¹ · Yu Deng¹

Received: 26 October 2018 / Accepted: 26 March 2019 / Published online: 11 April 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

With the increasing application of mobile devices, such as smart phones, the mobile ad hoc networks (i.e. MANETs) have been a hot research topic, in which how to effectively and efficiently search resources in MANETs attracts much attention. Due to peers' short wireless transmission range and strong mobility, the existing approaches suffer from the problem of lower search efficiency. This paper proposes a resource search strategy with the ant colony optimization, *tieSearch*, where we use the time-aware neighbor peer's availability and the time-aware neighbor peer's resource preferences as the pheromones to guide the ants to the holders of the requested resources under the finding that most peers' movements change regularly on a daily basis in the MANET environments. We detail the calculations of the abovementioned two types of pheromones, including their initial value settings, their increment updates and their attenuations. Also, we discuss the handling approach to alleviate the impact of peers' random churn on the resource search efficiency. The simulation results show that the *tieSearch* strategy outperforms other strategies in terms of the successful search rate, the traffic overhead and the search time.

Keywords Resource search · Peer's availability · Peer's resource preferences · Time-aware · Ant colony · MANET

1 Introduction

In recent years, the mobile networks especially the mobile ad hoc networks (i.e. MANETs) have attracted much attention with the rapid advancement of wireless communication techniques and the popular application of mobile devices, such as smart phones. A MANET is composed of the peers connected wirelessly without using any infrastructure and central servers [1–7]. In such a network environment, how to effectively and efficiently search resources has been an important and difficult research topic due to the fact that each peer has limited transmission range and strong mobility [8].

A lot of resource search strategies have been proposed in MANETs, which can be classified into four types, blind search [9–11], exact search [12, 13], heuristic search [5, 14, 15] and other types of search [4, 16–18]. The flooding and random walk approaches belong to the blind search. High traffic overhead is the main shortcoming of the flooding approach. Though the random walk approach can reduce the network

overhead by forwarding requests to the randomly selected neighbor peers, its search success rate cannot be guaranteed due to the randomness of selecting the path through which the requests are forwarded. The exact search approaches usually rely on the indices of resources or distributed hash table (DHT) [12, 13, 19]. How to manage the indices or how to implement the DHT in the highly dynamic MANETs is the main issue to be solved in such approaches. Moreover, it is difficult for an exact search approach to implement the key-based search process. The heuristic search is implemented based on the historical search records, which forwards the requests to the peers selected by using the peers' historical resource providing experience [5, 14, 15]. In addition to the abovementioned approaches, the search strategies adopting peers' clustering, resource replications and search path optimization were also proposed, expecting to improve resource search efficiency in MANETs.

As detailed in Section 2, the existing resource search strategies mentioned above have their inherent shortcomings in MANET environments. For example, the strategies using peers' clustering approach only took the peers' locations into account to design resource search algorithms, but paid less attention to the time factor, resulting in lower resource search efficiency in MANETs. This is because the existing studies [20–22] indicated that most peers'

✉ Xianfu Meng
xfmeng@dlut.edu.cn

¹ School of Computer Science and Technology, Dalian University of Technology, No. 2, Linggong Road, Dalian 116024, China

movements change regularly on a daily basis, which implies the time is the most important factor causing the dynamicity of a MANET. Considering the fact that each peer has limited transmission range in MANET, a peer's movement on a daily basis means the peer's one-hop neighbor peers within the peer's wireless transmission range would change over time. This is why the peers are highly dynamic and the existing resource search strategies could not adapt better to the MANET environments. Based on this analysis, we present the time-aware resource search strategy with the ant colony optimization in MANETs, *tieSearch* ("tie" is short for time-aware). Our main work is the following.

- (1) We analyze the characteristics of peer activities. We divide a peer's activities in MANET into two parts, one is the peer's online intervals and another is the peer's resource preferences. We use a peer's availability in a specific time interval to quantify the peer's online probability in the specific time interval. Combined with the calculations of the peer's resource preferences, the peer's activities related to the resource search can be summarized as the time-aware availability and the time-aware resource preferences.
- (2) We establish a time-aware ant colony model for resource search in MANETs. In our time-aware ant colony model, we use the abovementioned peers' time-aware availability and time-aware resource preferences as the pheromones, by which an ant with the request message can be guided to the neighbor peer with higher availability and higher probability that the peer holds the requested resource in the current time interval.
- (3) We design an approach to tackle the peers' random movements. Though we employ the time-aware approach to reduce the impact of peers' strong mobility on the resource search efficiency based on the finding that a peer's movements change regularly on a daily basis, we also present an approach to handle peers' random movements so as to improve the resource search efficiency in MANETs.

The remainder of the paper is organized as follows. Section 2 presents the related work on the resource search in MANETs. Section 3 discusses the calculations of the time-aware neighbor peers' availability and resource preferences for laying the foundation of establishing an ant colony model to improve resource search efficiency. In Section 4, we detail the time-aware resource search approach with the ant colony optimization in MANETs, where we use neighbor peers' time-aware availability and resource preferences as the pheromones. The simulations and result analyses are provided in Section 5. In Section 6, we conclude the paper and supply our future research focus.

2 Related work

In a MANET, each peer has limited transmission range and strong mobility. In such an environment, how to grasp a peer's one-hop neighbor peers in the peer's wireless transmission range is a crucial issue in designing a resource search strategy in MANET, since knowing a peer's one hop neighbor peers can help the peer select a suitable neighbor peer to which the resource request is sent or forwarded. Till now, a lot of resource search approaches have been proposed in MANETs, which can be classified into 4 types as detailed below.

2.1 Blind search

The flooding [9] and random walk [10] approaches are the representatives of blind search. In the flooding approach, when a peer receives a request, the peer first checks whether or not it holds the requested resource. If so, the peer returns the resource or its information to the requester; otherwise, the request is forwarded to all the peer's neighbors until TTL becomes zero. Different from the flooding approach, the random walk approach forwards the request to the randomly selected neighbors. Hence, the number of the selected neighbors is an important factor affecting the search success rate. The expanding ring search [11] is designed based on the random walk approach. It employs a variable TTL instead of the fixed one used in the random walk approach. When a search failed to get its requested resource within the current TTL, this approach increases the value of TTL to widen the search scope. High traffic overhead and low search efficiency are the shortcomings of such approaches.

2.2 Exact search

Exact search is also called the exact matching search. This type of resource search usually uses the indices of resources to rapidly locate the holders of the requested resources. Two ways are employed to manage the indices, the centralized index server [12] or the distributed hash table (DHT) [19]. Since using a central server to manage resources' indices suffers from the problem of single point of failure, the DHT approach is usually used, which distributes a resource's index on the peer determined by the hashing value of the resource. Thus, a balanced indices distribution can be achieved. However, in a highly dynamic MANET, the maintenance cost of DHT is high. To mitigate the cost, work in [13] adopts the clustering method. It divides an overall network into several smaller ones, each of which uses DHT to manage the indices of resources held by the peers in it. Though such strategy could to some degree reduce the DHT maintenance cost, the cluster maintenance cost is increased. In addition, it is not difficult to image that using DHT to manage resources' indices is not the best choice for resource search in a highly dynamic MANET.

2.3 Heuristic search

Using a peer's historical search information to guide the peer to send or forward its created or received requests is the common feature of the heuristic search. Here, how to use the historical search information is an issue to be considered. Usually, a peer can abstract or calculate its neighbor peer's trust, the quality of resources provided by the neighbor peer and the neighbor peer's resource interests or preferences from the peer's managed historical search information. Work [14] takes a neighbor peer's trust as a factor to send or forward requests. It divides a neighbor peer's trust into two parts, the acquaintance trust and the similarity trust. A peer quantifies its neighbor peer's acquaintance trust using the number of interactions with the neighbor peer, and a peer calculates its neighbor peer's similarity trust by using the similarity of the resources owned by the peer and provided by the neighbor peer. Since only relying on the neighbor peers' trust to forward requests has limited effect on improving resource search efficiency, works in [5, 15] add some other factors to determine the request forwarding route. For example, work [5] takes a neighbor peer's mobility into consideration to forward request, and a peer's request would be first sent to the neighbor peer that would move out of the peer's wireless transmission range soon. Similarly, work in [15] takes a neighbor peer's selfishness and preferences into account to select the request forwarding path. Though the existing heuristic search approaches could to some extent improve resource search efficiency, the improvement is limited due to the fact that they did not deeply consider the impact of strong peers' mobility on the search performance in MANETs.

2.4 Other types of search

Researchers also use other types of strategies to improve resource search efficiency in MANETs, such as peer clustering, resource replication and search path optimization. The clustering methods improve resource search efficiency by clustering peers based on locations or interests [17, 23]. A request is sent or forwarded to the peers in the local cluster. Only when the local search failed would the search be performed in other clusters. Since the cluster maintenance is conducted by periodically sending detection messages, this type of strategy is suitable for the MANET where peers have lower mobility.

2P-Lookup [18] improves resource search efficiency by constructing a P2P overlay based on peers' physical proximity over MANETs, and uses resource popularity-biased random walk to send search requests. Here, a peer's proximity is detected by broadcasting the connection messages and a resource's popularity is computed by combining the local knowledge and the global knowledge about the resource. This work employs the resource popularity to calculate the number of walkers and their TTL which are used in the

random walk approach. However, it suffers from the shortcomings that maintaining a proximity-based P2P overlay is costly in the highly dynamic MANETs and the improvement of the resource search efficiency is limited without taking the time factor into account to determine the walkers under the finding that most peers' movements change regularly on a daily basis in MANETs.

The resource replication-based search strategies improve search efficiency by replicating a resource to multiple peers. Considering the strong mobility of peers, the replicas of resources are usually distributed on the peers with lower mobility. E. Atsan et al. [24] proposed a resource replication approach based on the so-called connected dominating set. Here, the connected dominating set refers to such a subset of peers that every peer is either in the subset or only one hop away from the subset. M. Pushpalatha et al. [25] also replicates resources to the dominating set. They employ two phases to complete a resource replication. The first phase is used to identify and minimize the dominating set on which the resources are replicated. In the second phase, a steady peer is selected and identified, on which the replica of the resource whose holder would move out of a given range is relocated. To complete the operations of the two phases, the authors proposed several relevant algorithms, such as the dynamic replica allocation algorithm, the mobility prediction algorithm and the replica relocation algorithm.

The ant colony algorithm is a simulated evolutionary computation inspired by the foraging behavior of ant colonies. According to the studies, the ant colony algorithm is suitable for searching resources in the distributed environments [26, 27]. To improve resource search efficiency in MANETs, work in [26] employs the ant colony algorithm to select the path through which the requests are forwarded. To tackle peers' mobility, each peer periodically broadcasts *Hello* messages to its neighbor peers. If a broken path (neighbor) is found, it chooses an alternative path based on the pheromones. G. Singh et al. [27] takes the orientation as a factor to calculate the pheromone. If an intermediate peer through which the ant passes and the target peer have different orientations, then the ant would remain fewer pheromone on the intermediate peer.

The abovementioned existing resource search approaches suffer from a common shortcoming that they are unable to effectively and efficiently adapt to the highly dynamic MANET environments due to the fact that none of those approaches makes use of the finding that a peer's movements usually change regularly on a daily basis [20–22].

3 Time-aware neighbor peers' availability and resource preferences

As mentioned above, the existing researches indicated that a peer's movements in a MANET regularly change on a daily

basis [20–22]. Such regular movements of a peer are called the peer's movement pattern. Obviously, peers' movement patterns determine the dynamic characteristics of a MANET, and thus determine a peer's neighbor peers able to be directly interacted with in different time intervals. For the resource search in MANET, it is important for a peer to have a good grasp of its neighbor peers in different time intervals due to the fact that the peer can directly send or forward requests to its online neighbor peers without using the flooding approach. Furthermore, such time-aware neighbor peers of a peer also determine the time-aware resource preferences of the peer, since the existing researches indicated that each peer in a MANET has its own interests and the resources a peer can provide are heavily related to the peer's interests [28–30]. Therefore, in order to improve resource search efficiency in MANETs, it is important to first have a grasp of the time feature of a peer's activities, and then achieve the peer's time-aware neighbor peers and their resource preferences.

3.1 Brief introduction to the time feature of a peer's activities

A lot of researches [20–22] focused on the time feature of a peer's activities in MANETs, in which work in [20] analyzed the activities of more than 7 thousands of peers in mobile network environments by visiting their system log files, telephoning records and interactive data packages, which were produced by the peers over 17 weeks. The authors found that most mobile devices stayed in some fixed areas for longer. 95.1% of the users of mobile devices have their fixed active areas (i.e. the so-called home locations), and they spent 98.7% of their time there. In such fixed areas, the mobile devices tended to be online persistently. Furthermore, by analyzing a large amount of mobile peers' interactive data, researchers found that most mobile devices move regularly between their fixed areas on a daily or weekly basis [20–22]. In other words, each peer has its own movement patterns which change on a daily basis.

The abovementioned finding gives us a hint that we can improve resource search efficiency in MANETs by making full use of the peers' movement patterns, since a peer's movement patterns can help us have a good grasp of the peer's neighbor peers and their resource preferences in a specific time interval. Here, a peer's time-aware neighbor peers can be determined by examining the neighbor peers' availability in the given time intervals, and a neighbor peer's resource preference can be determined by the successful search rate of a specific resource provided by the neighbor peer.

3.2 Calculations of the time-aware neighbor peers' availability

As mentioned above, a good grasp of a peer's time-aware neighbor peers can help improve resource search efficiency

in MANET environments under the finding that a peer's movements regularly change on a daily basis. To obtain a peer's time-aware neighbor peers, we can calculate the neighbor peers' availability of the peer in each time interval during a day. To this end, each peer needs managing its resource search information, i.e. the transaction information.

A peer's transaction information is shown in Table 1, where *NeighborID* is the neighbor peer's ID from which the current peer received or routed the responding message in the transaction. *Time* is the time at which the current peer received or routed the responding message. *ResourceVector* is used to represent the requested resource, which consists of one or more resource keys used to indicate the features of the requested resource. *Path* saves the information of the path from the requester to the provider. In *Path*, the first peer is the requester and the last peer is the provider.

The initial transaction information of a peer is achieved by flooding requests, since in this phase we have no useful information to guide us to improve resource search efficiency. However, when a peer accumulated enough transaction data, our strategy can be applied to improve resource search efficiency in MANETs.

Based on the accumulated transaction data of a peer as shown in Table 1, we first equally divide a day into n time intervals, $[t_1, t_2][t_2, t_3] \dots [t_n, t_{n+1}]$, where $\Delta t_i = [t_{i+1} - t_i]$. Then, for that peer, we calculate its daily transactional times (represented by $S_{\Delta t_j}^d(N_i)$) with any neighbor peer N_i during any time interval Δt_j , its daily total transactional times (denoted by $S_{\Delta t_j}^d$) with all its neighbor peers during any time interval Δt_j , and its total transactional times (denoted by $Sum_{\Delta t_j}(N_i)$) with any neighbor peer N_i in m days, based on the transactional data listed in Table 1. We save the calculated results in the form of Table 2.

In Table 2, $S_{\Delta t_j}^d(N_i)$, $S_{\Delta t_j}^d$ and $Sum_{\Delta t_j}(N_i)$ are calculated with Formulae (1)–(3), respectively.

$$S_{\Delta t_j}^d(N_i) = \text{countif}(R \in \text{Table 1}, R.Time.d = d \& \& R.Time.t \in \Delta t_j \& \& R.NeighborID = N_i) \quad (1)$$

$$S_{\Delta t_j}^d = \sum_{i=1}^k S_{\Delta t_j}^d(N_i) \quad d = 1-m \quad (2)$$

$$Sum_{\Delta t_j}(N_i) = \sum_{d=1}^m S_{\Delta t_j}^d(N_i) \quad i = 1-k \quad (3)$$

where R is a record in Table 1; $R.Time.d$ represents the day item of the *Time* field in record R ; $R.Time.t$ stands for the time item of the *Time* field in record R .

Table 1 Structure of a peer's transaction information

NeighborID	Time	ResourceVector	Path
------------	------	----------------	------

Table 2 Transactional times in the time interval Δt_j

	1st day	2nd day	3rd day	...	m^{th} day	
N_1	$s_{\Delta t_j}^1(N_1)$	$s_{\Delta t_j}^2(N_1)$	$s_{\Delta t_j}^3(N_1)$...	$s_{\Delta t_j}^m(N_1)$	$sum_{\Delta t_j}(N_1)$
N_2	$s_{\Delta t_j}^1(N_2)$	$s_{\Delta t_j}^2(N_2)$	$s_{\Delta t_j}^3(N_2)$...	$s_{\Delta t_j}^m(N_2)$	$sum_{\Delta t_j}(N_2)$
N_3	$s_{\Delta t_j}^1(N_3)$	$s_{\Delta t_j}^2(N_3)$	$s_{\Delta t_j}^3(N_3)$...	$s_{\Delta t_j}^m(N_3)$	$sum_{\Delta t_j}(N_3)$
...
N_k	$s_{\Delta t_j}^1(N_k)$	$s_{\Delta t_j}^2(N_k)$	$s_{\Delta t_j}^3(N_k)$...	$s_{\Delta t_j}^m(N_k)$	$sum_{\Delta t_j}(N_k)$
	$s_{\Delta t_j}^1$	$s_{\Delta t_j}^2$	$s_{\Delta t_j}^3$...	$s_{\Delta t_j}^m$	

Based on the data in Table 2, the peer further calculates its daily transactional probability with any neighbor peer N_i during any time interval Δt_j using Formula (4) and the average transactional probability with any neighbor peer N_i during any time interval Δt_j in m days using Formula (5).

$$p_{\Delta t_j}^d(N_i) = \frac{S_{\Delta t_j}^d(N_i)}{S_{\Delta t_j}^d} \quad (4)$$

$$\overline{p_{\Delta t_j}(N_i)} = \frac{1}{m} \sum_{d=1}^m p_{\Delta t_j}^d(N_i) \quad (5)$$

From Formula (5), we see that the value of $\overline{p_{\Delta t_j}(N_i)}$ is in $[0, 1]$, and the higher the value of $\overline{p_{\Delta t_j}(N_i)}$ is, the higher the probability that neighbor peer N_i is online in the time interval of Δt_j is, and vice versa.

Since the standard deviation can help us grasp the fluctuation of $\overline{p_{\Delta t_j}(N_i)}$, we calculate the standard deviation of any neighbor peer N_i 's daily transactional probability using Formula (6).

$$SD_{\Delta t_j}(N_i) = \sqrt{\frac{1}{m} \sum_{d=1}^m (p_{\Delta t_j}^d(N_i) - \overline{p_{\Delta t_j}(N_i)})^2} \quad (6)$$

We know that the less the value of $SD_{\Delta t_j}(N_i)$ is, the less the fluctuation of the transactional probability with neighbor peer N_i in the time interval of Δt_j every day is. Hence, we use Formula (7) to calculate and normalize the availability of neighbor peer N_i in the time interval of Δt_j .

$$U_{\Delta t_j}(N_i) = e^{-\delta_{\Delta t_j}(N_i)} \quad (7)$$

$$\delta_{\Delta t_j}(N_i) = \frac{SD_{\Delta t_j}(N_i)}{\overline{p_{\Delta t_j}(N_i)}} \times \frac{1}{Sum_{\Delta t_j}(N_i)} \quad (8)$$

From the definition of Formula (7), we see that the higher the value of $U_{\Delta t_j}(N_i)$ is, the higher the availability of neighbor peer N_i in the time interval of Δt_j is, and vice versa. This means that the value of $U_{\Delta t_j}(N_i)$ can be used to determine whether neighbor peer N_i is online or not in the time interval of Δt_j and help us grasp the available neighbor peers of a peer in the time interval of Δt_j , and thus a reachable path can be guaranteed

with higher probability for the peer to send or forward a request for a resource in the time interval of Δt_j under the environment of a MANET that each peer has limited wireless transmission range.

We take the following example to show the process of calculating the neighbor peers' availability for a peer in a specific time interval. Figure 1 shows peer A's five neighbor peers, i.e. B, C, D, E and F. Table 3 lists peer A's transactional times with the five neighbor peers during the time interval of Δt_j in five days. Then, we calculate peer A's transactional probability with each neighbor peer in the time interval Δt_j every day, the average transactional probability with each neighbor peer in five days, the standard deviation of the transactional probability and each neighbor peer's availability in the time interval of Δt_j using Formula (1)–(7). The results are listed in Table 4, where *A.T.P* represents the average transactional probability, *S.D* stands for the standard deviation and *AVL* is the peers' availability. From Table 4, it is not difficult to understand that among all the neighbor peers of peer A, neighbor peer D has the highest availability and neighbor peer C has the lowest availability in the time interval of Δt_j .

The above calculated time-aware neighbor peers' availability would be used to implement our resource search strategy with the ant colony optimization in MANETs, where a neighbor peer's time-aware availability is taken to be the initial value of the neighbor peer's availability pheromone, as detailed in Section 4.

3.3 Calculations of the time-aware neighbor peers' resource preferences

In the previous section, we detailed how to grasp a peer's neighbor peers in any time interval by calculating the neighbor peers' availability. When a peer wants to send a request for a

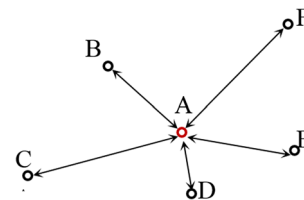


Fig. 1 Current peer A's Neighbors

Table 3 Neighbors' transactional times with A

	1st	2nd	3rd	4th	5th	Total
B	2	3	3	2	4	14
C	2	0	3	0	0	5
D	3	5	4	4	5	21
E	3	0	2	2	3	10
F	0	0	0	9	0	9
	10	8	12	7	12	

resource to its neighbor peer, it should select the neighbor peer whose availability in the current time interval is higher to receive the request, since a neighbor peer with higher availability in the current time interval means the probability that it is both being online and being within that peer's wireless transmission range is high, and thus reducing both the retransmission probability and the network overhead. However, only relying on sending or forwarding requests to the neighbor peer with high availability in the current time interval cannot guarantee the significant increase of resource search efficiency, since each peer in a MANET has its own interests [28–30]. Therefore, to improve resource search efficiency (i.e. increasing successful search rate and lowering network overhead), we should also take the neighbor peers' resource preferences (i.e. interests) into account to select the neighbor peer to which the request should be sent or forwarded.

Existing researches indicated that any peer in a MANET has its own interests [28–30], and the resources a peer is able to provide are highly related to the peer's interests. Also, any resource can be represented by a group of keys (i.e. the resource vector consisting of interest keys or resource keys), and a peer can create a request for a resource by setting the resource vector to the request. Here, a peer's interests can be grasped by analyzing the peer's resource preferences.

A peer's resource preferences can be achieved by analyzing the resource search responding messages (i.e. the transaction information as listed in Table 1) received or routed by the peer. For this, we classify the responding messages a peer received into two kinds, the messages of responding to the requests issued by the peer's neighbor peers and the messages of responding to the requests routed

by the peer's neighbor peers. Here, we name the first kind of responding message the direct responding message, and call the resource keys included in the first kind of responding message the direct resource preferences. Similarly, we name the second kind of responding message the indirect responding message, and call the resource keys included in the second kind of responding message the indirect resource preferences. Apparently, a neighbor peer's resource preferences should be composed of its direct resource preferences and indirect resource preferences. This is because the direct resource preferences indicate the type of resources the neighbor peer is interested in, and the indirect resource preferences represent the type of resources the neighbor peer is interested in with high probability due to the fact that any request for a resource should be routed by the peers whose interests are similar to those indicated by the request. Based on this consideration, we calculate a neighbor peer's direct resource preferences and indirect resource preferences as follows.

We respectively calculate a peer's neighbor peers' direct resource preferences and indirect resource preferences using the peer's transaction data of m days listed in Table 1.

We use $direct_{\Delta t_j}^{key}(N_i)$ and $indirect_{\Delta t_j}^{key}(N_i)$ to respectively represent neighbor peer N_i 's direct resource preference and indirect resource preference for the resource indicated by key in the time interval of Δt_j . Then, $direct_{\Delta t_j}^{key}(N_i)$ can be represented by the total number of requests for the resource indicated by key issued by neighbor peer N_i in the time interval of Δt_j , and $indirect_{\Delta t_j}^{key}(N_i) = \sum^{\perp} ?Number\ of\ hops\ from\ each\ Requester\ to\ the\ current\ peer$, where the *Requester* sends the requests for the resources indicated by key routed by neighbor peer N_i in the time interval of Δt_j . From the calculation of $indirect_{\Delta t_j}^{key}(N_i)$, we see that the higher the number of hops that a request with resource key comes from the requester via neighbor peer N_i to the current peer in the time interval of Δt_j is, the less the impact of the requester's request on neighbor peer N_i 's indirect resource preferences is.

Based on the calculations of $direct_{\Delta t_j}^{key}(N_i)$ and $indirect_{\Delta t_j}^{key}(N_i)$, we can achieve neighbor peer N_i 's resource preference for the resource indicated by key in the time interval of Δt_j using Formula (9).

Table 4 Calculated neighbor peers' availability

	1st	2nd	3rd	4th	5th	A.T.P	S.D	$\delta_{\Delta t_j}(N_i)$	AVL
B	0.2	0.375	0.25	0.117	0.333	0.255	0.0923	0.0259	0.974
C	0.2	0	0.25	0	0	0.09	0.1114	0.2476	0.781
D	0.3	0.625	0.333	0.235	0.417	0.382	0.1349	0.0186	0.981
E	0.3	0	0.167	0.117	0.25	0.1668	0.1048	0.0628	0.939
F	0	0	0	0.531	0	0.1062	0.2124	0.2222	0.801

$$r_{\Delta t_j}^{key}(N_i) = direct_{\Delta t_j}^{key}(N_i) + indirect_{\Delta t_j}^{key}(N_i) \quad (9)$$

Formula (9) calculates neighbor peer N_i 's resource preference for the resource indicated by key in the time interval of Δt_j . Similarly, we can use Formula (9) to calculate neighbor peer N_i 's resource preference for the resource indicated by any key in the time interval of Δt_j . Therefore, any neighbor peer N_i 's preferences for the resources indicated by different keys in the time interval of Δt_j can be represented by a vector, as shown in Formula (10).

$$R_{\Delta t_j}(N_i) = (r_{\Delta t_j}^{key_1}(N_i), r_{\Delta t_j}^{key_2}(N_i), \dots) \quad (10)$$

We take the following example to show the process of calculating neighbor peers' resource preferences in a specific time interval. Figure 2 illustrates the relationships of current peer A and other peers, B, C, D, E, F and G. Table 5 lists the responding messages current peer A received, which is abstracted from Table 1. We use Formula (9) to calculate the resource preferences of neighbor peers B, E and F for peer A in the time interval of (14:00–15:00). The results are shown in Table 6.

When a peer just joined-in the MANET, it can search the resources by flooding its requests to accumulate its transaction data. When the peer accumulated enough transaction data, we can start calculating the time-aware neighbor peers' availability and resource preferences for the peer. In a realistic situation, according to the finding that peer's movements change regularly on a daily or weekly basis, we can start calculating the time-aware neighbor peers' availability and resource preferences after the peer has joined the MANET for a week. Namely, we set the minimum value of m to 7.

The above calculated time-aware neighbor peers' resource preferences would be used to implement our resource search strategy with the ant colony optimization in MANET, where a neighbor peer's time-aware resource preference is taken to be the initial value of the neighbor peer's resource preference pheromone, as detailed in Section 4.

4 Time-aware resource search approach with the ant colony optimization

Existing researches indicated that the ant colony algorithm has the ability to find optimal solutions and is suitable for being

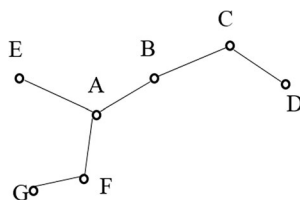


Fig. 2 Relationships of current peer A and other peers

Table 5 Responding messages received by current peer A

NeighborID	Time	ResourceVector	Path
B	14:12	(2017,USA,Film)	G-F-A-B-C-D
E	14:27	(2017,CHN,Film)	C-B-A-E
F	14:48	(2018,CHN,Variety)	E-A-F-G
B	14:51	(2018,USA,Variety)	F-A-B

used in distributed computing [26, 27]. Due to the fact that the ant colony algorithm can guide an ant to the food source based on the pheromones without traversing all the peers, it is suitable for solving the resource search problem in MANETs.

4.1 Related definitions

4.1.1 The structure of an ant

The structure of an ant is shown in Table 7, where *Header* is composed of the requester's ID and the ant's ID. If two ants have the same header, then we consider the two ants as the same one; *ResourceVector* is used to represent the requested resource, which consists of one or more resource keys used to indicate the features of the requested resource; *State* stands for the state the ant is in, including *Active* and *Death*. An ant in the *Active* state means it can walk, query and return. An ant in the *Death* state means the ant has already finished its task. The *Path* field records the information of path the ant passes through, which would be used in the ant returning process. The first peer ID in *Path* is the requester's ID and the last peer ID in *Path* is the provider's ID. *TTL* stands for the time to live.

4.1.2 The rules of an ant foraging

In a resource search strategy with the ant colony optimization, it is important to determine the ant foraging rules, since the ant foraging rules are directly related to the resource search efficiency and effectiveness. In our strategy, an ant finds its food according to the following rules.

1. Walking rule. An ant always walks towards the peer with higher pheromones. If an ant cannot detect the existence of pheromones, it randomly selects a path to walk.
2. Re-walking rule. When an ant is unable to reach the peer selected with the *Walking* rule (e.g., due to churn), it would use the *Walking* rule again to select a new path to walk.
3. Pheromone setting rule. The nearer the ant to the food source is, the higher the pheromones it should remain, and vice versa.

Table 6 Neighbor peers’ resource preferences of peer A

NeighborID	Resource preference vector
B	$R_{14:00-15:00}(B) = (r_{14:00-15:00}^{2017}(B) = 0.33, r_{14:00-15:00}^{2018}(B) = 1, r_{14:00-15:00}^{USA}(B) = 1.33, r_{14:00-15:00}^{Variety}(B) = 1, r_{14:00-15:00}^{Film}(B) = 0.33)$
E	$R_{14:00-15:00}(E) = (r_{14:00-15:00}^{2017}(E) = 1, r_{14:00-15:00}^{CHN}(E) = 1, r_{14:00-15:00}^{Film}(E) = 1)$
F	$R_{14:00-15:00}(F) = (r_{14:00-15:00}^{2018}(F) = 0.5, r_{14:00-15:00}^{CHN}(F) = 0.5, r_{14:00-15:00}^{Variety}(F) = 0.5)$

4.1.3 The foraging process of an ant

The foraging process of an ant consists of three actions, walking, querying and returning. The *Walking* action means the ant is walking forward to find the food source according to the *Walking* or *Re-walking* rule. The *Querying* action implies the ant is foraging locally when it walked to a peer. When an ant is taking the *Returning* action, it would be returning to its nest (the peer that issued the ant, i.e. the requester) and updating the pheromones along the path (indicated by the *Path* field in the ant structure) it came according to the *Pheromone setting* rule.

4.1.4 Classification of the pheromones

As mentioned in the end of Section 3, our strategy uses two types of pheromones, one is the pheromone of neighbor peer’s availability, and another is the pheromone of neighbor peer’s resource preference. The two types of pheromones respectively reflect the key points a resource search strategy should be seriously considered in MANETs, i.e. how to handle the network dynamicity and how to find the holder of the requested resource. From the descriptions in Sections 3.2 and 3.3, we know that the higher neighbor peer’s availability in a time interval can guarantee that the probability that the neighbor peer is being online in the time interval is high, and the higher neighbor peer’s resource preference for a resource represented by a resource key means that the probability that the neighbor peer holds the resource is high. Therefore, we employ the two types of pheromones to implement our ant colony-based resource search strategy in MANETs.

4.2 Calculation and update of the pheromones

In an ant colony algorithm, it is important to design an effective and efficient pheromone update approach, including both the accumulation and attenuation of pheromones. Generally speaking, we employ Formula (11) to update the pheromones.

$$\tau(t_{new}) = \rho \cdot \tau(t_{old}) + \Delta\tau \tag{11}$$

Table 7 Structure of an ant

Header	ResourceVector	State	Path	TTL
--------	----------------	-------	------	-----

where $\tau(t_{old})$ represents the original value of the pheromone updated in the time of t_{old} , and $\tau(t_{new})$ stands for the newly updated value of the pheromone in the time of t_{new} . $\Delta\tau$ is the pheromone’s increment in the time interval of Δt_j . Note that Δt_j is determined by t_{new} , i.e. Δt_j is the time interval that t_{new} is in. $\rho \in (0, 1)$ is the attenuation factor of the pheromone in the time interval of t_{old} to t_{new} , which is calculated with Formula (12).

$$\rho = e^{-(t_{new} - t_{old})} \tag{12}$$

where t_{old} and t_{new} are in hours.

Since we use two types of pheromones, the neighbor peer’s availability and the neighbor peer’s resource preferences, we respectively detail the calculations of the two types of pheromones in the following.

(1) Update of a neighbor peer’s availability pheromone

Based on Formula (11), we use Formula (13) to update a neighbor peer’s availability pheromone.

$$U\tau_{\Delta t_j}^{N_i}(t_{new}) = \rho \cdot U\tau_{\Delta t_j}^{N_i}(t_{old}) + \Delta U\tau_{\Delta t_j}^{N_i} \tag{13}$$

where $U\tau_{\Delta t_j}^{N_i}(t_{old})$ is the original value of neighbor peer N_i ’s availability pheromone updated in the time of t_{old} . Its initial value is set to that calculated in Section 3.2. ρ is the attenuation factor calculated using Formula (12). $\Delta U\tau_{\Delta t_j}^{N_i}$ is the increment of neighbor peer N_i ’s availability pheromone in the time interval of Δt_j , which is calculated using Formula (14). Note that Δt_j is the time interval t_{new} is in.

$$\Delta U\tau_{\Delta t_j}^{N_i} = \frac{1}{S_{\Delta t_j}} \tag{14}$$

where $S_{\Delta t_j}$ represents the total number of transactions established between the current peer and its neighbor peers in the time interval of Δt_j .

(2) Update of a neighbor peer’s resource preference pheromone

After a successful search for a resource, the resource preference pheromone of the neighbor peer through which the current peer received the responding message should be updated. Note that a resource can be represented by multiple resource keys, and thus we should update the pheromone of

the neighbor peer's resource preference for each resource key. Remember that a neighbor peer's resource preference pheromone is also time-aware, which is similar to the pheromone of the neighbor peer's availability.

Based on Formula (11), we employ Formula (15) to update neighbor peer N_i 's resource preference pheromone for each resource key included in *ResourceVector* of the ant.

$$S\tau_{\Delta t_j}^{N_i}(t_{new}) = \rho \cdot S\tau_{\Delta t_j}^{N_i}(t_{old}) + \Delta S\tau_{\Delta t_j}^{N_i} \quad (15)$$

where $S\tau_{\Delta t_j}^{N_i}(t_{old})$ is the original value of neighbor peer N_i 's resource preference pheromone for a corresponding resource key updated in the time of t_{old} . Its initial value is set to $r_{\Delta t_j}^{key}(N_i)$ calculated in Section 3.3. ρ is the attenuation factor calculated using Formula (12). $\Delta S\tau_{\Delta t_j}^{N_i}$ is the pheromone's increment of neighbor peer N_i 's resource preference for a corresponding resource key in the time interval of Δt_j , which is calculated using Formula (16). Note that Δt_j is the time interval t_{new} is in.

$$\Delta S\tau_{\Delta t_j}^{N_i} = \frac{1}{L} \quad (16)$$

where L is the number of hops from the current peer to the provider (i.e. the peer holding the requested resource). Here, that we use Formula (16) to calculate the increment of a neighbor peer's resource preference pheromone is based on the *Pheromone setting* rule given in Section 4.1.

Note that if the range of the time at which a request is sent and the time when the message of responding to the request is received is across multiple given time intervals (e.g. $\Delta t_i, \Delta t_{i+1}, \dots$), then all the neighbor peers' pheromones in all the crossed time intervals should be updated, so as to reflect the time-aware pheromones' characteristics.

4.3 Resource search strategy with the ant colony optimization

In Section 4.2, we described the calculation and update of the time-aware neighbor peer's availability pheromone and the time-aware neighbor peer's resource preference pheromone. The two types of pheromones lay the foundation for establishing the resource search strategy with the ant colony optimization in MANETs. In an ant colony model-based resource search strategy, the process of searching a resource is the same as the process of an ant foraging. Here, we divide an ant foraging process into three phases, initializing, searching and returning. In the initializing phase, the main work is to create an ant with the resource search request. In the searching phase, the ant needs searching the requested resource locally or selecting a neighbor peer to pass through when failed to find out the requested resource. When an ant finds out the requested resource, it enters its returning phase. In this phase, the ant returns to its nest along the path it came and updates the

pheromones of the peers in the path. We will detail the three phases in the following.

4.3.1 Initializing phase

In the initializing phase, the requester should create an ant with the structure as defined in Table 7. This procedure is listed in the following.

CreateAnt(*antID*, *requesterID*, *ResourceVector*, *TTL*)

- Step 1: Create a new *ant*
- Step 2: *ant.Header* = (*requesterID*, *antID*)
- Step 3: *ant.ResourceVector* = *ResourceVector*
- Step 4: *ant.State* = *Active*
- Step 5: *ant.Path* = *requesterID*
- Step 6: *ant.TTL* = *TTL*
- Step 7: return *ant*

In the *CreateAnt* procedure, the parameter of *ResourceVector* indicates what resource wants to be searched by the created ant, which consists of one or more resource keys representing the requested resource.

4.3.2 Searching phase

After the initializing phase completed, the created ant enters its searching phase. In this phase, two things should be performed, one is to search the requested resource locally and the two is to select a neighbor peer through which the ant walks forward, if the local search failed.

(1) Local search

When an ant arrives at a peer (i.e. current peer), it first checks whether the peer has been walked through already. If so (meaning that the ant enters a loop path), the ant enters its *Death* state. Otherwise, the ant locally searches the requested resource indicated by the *ResourceVector* the ant brings. The local search procedure is shown as follows.

LocalSearch(*ant*, *currentPeer*)

- Step 1: If the *ant* has walked *currentPeer* already, the *ant* enters its *Death* state;
- Step 2: The *ant* checks whether there exists the resource represented by *ant.ResourceVector* in *currentPeer*. If so, the *ant* enters its returning phase;
- Step 3: if $-ant.TTL > 0$, then the *ant* sets its *Path* information and walks forward through *currentPeer*'s neighbor peer selected using the procedure of *SelectNeighbor*. Otherwise, the *ant* enters its *Death* state.

In the procedure of *LocalSearch*, *currentPeer* is the peer the *ant* just arrived at and *ant* is the ant arriving at *currentPeer*.

(2) Selection of the neighbor peer

According to the foraging rule given in Section 4.1, the ant with the request message should walk forward through the path (i.e. the neighbor peer) with higher pheromones. Considering the fact that we employ two types of the time-aware pheromones in our resource search strategy, the neighbor peer's availability pheromone and the neighbor peer's resource preference pheromone, we adopt the following procedure to choose a neighbor peer for the ant to walk through.

SelectNeighbor(ant, currentPeer)

- Step 1: Calculate the average value of the pheromones of all the resource keys the *ant.ResourceVector* owns for each neighbor peer across the current time interval, represented by *AverPrefer*;
- Step 2: Select the neighbor peer whose total resource preference pheromone of all the above mentioned resource keys is not less than *AverPrefer* and its availability pheromone is the highest in the current time interval, if exists. Otherwise, go to *Step 3*;
- Step 3: Select the neighbor peer that has the resource pheromones of the above mentioned resource keys and whose availability pheromone is the highest in the current time interval, if exists. Otherwise, go to *Step 4*;
- Step 4: Select the neighbor peer which has the highest availability pheromone in the current time interval.

If a neighbor peer is selected according to the procedure of *SelectNeighbor*, then the ant would walk to the selected neighbor peer as shown in the procedure of *LocalSearch*; or else, the flooding approach is used.

4.3.3 Returning phase

As mentioned in the procedure of *LocalSearch*, when an ant finds out its requested resource, it should bring the requested resource or the responding message to return back to its nest along the path it came. In the returning process, the ant should also update the abovementioned two types of pheromones for the follow-up foraging ants. The detailed procedure is shown in the following.

4.4 Approach to handle random churn problem

In Section 4.3, we described our resource search strategy with the ant colony optimization, where we did not take random

churn problem into account. Though the time-aware pheromones can tackle the churn problem caused by peers' movements, we should also consider the unpredictable and random churn caused by peer's failure or irregular movements in an ant returning phase, since such situation would probably guide an ant to the peer that is not within the current peer's wireless transmission range and thus the relevant pheromones cannot be correctly updated.

Figure 3 illustrates the abovementioned case. An ant is returning to its nest along the path of peers of A-B-C-E indicated by the ant's *Path*. In this situation, if an intermediate peer *B* failed, then the ant should select another path to walk back, so as to update the path's pheromones. We adopt the following approach to handle a peer's failure in an ant returning phase.

First, peer *A* checks whether peer *C* is in its table of the time-aware neighbor peer list. If so, the ant directly walks to peer *C* without through the path of peer *B*. If not so, peer *A* sends a message to its neighbor peers for checking whether peer *C* is in their time-aware neighbor peer lists. If peer *A*'s neighbor peer, say peer *D*, has peer *C* in its neighbor peer list, then peer *A* sends the ant to peer *D*, for the reason that peer *D* can further send the ant to peer *C*, as illustrated in Fig. 3. In this situation, we update the ant's *Path* of A-B-C to A-D-C, and update the time-aware pheromones in the corresponding peers. If none of peer *A*'s neighbor peers has peer *C* in its neighbor peer list, then we randomly select a time-aware neighbor peer for the ant to walk back.

5 Simulations and results

This section examines the performance of the *tieSearch* strategy focusing on the efficiency and effectiveness in comparison with the flooding approach [9], *2P-Lookup* [18] and *TOP* [5]. The *TOP* strategy uses peers' trust to select a neighbor peer to which the request is sent. Suppose peer *A* has previously interacted with peer *B*. Peer *C* would ask peer *A* about peer *B*'s trustworthiness when it wants to send a request to peer *B*, as it does not have interaction history with peer *B*. The strategy suffers from two shortcomings. One is how to perceive which peer has the trustworthiness of another peer we want to know. Another is that it does not take peers' preferences to select the peer to which the request is sent. Since the standard flooding approach propagates too many messages and thus wastes both precious and limited wireless bandwidth,

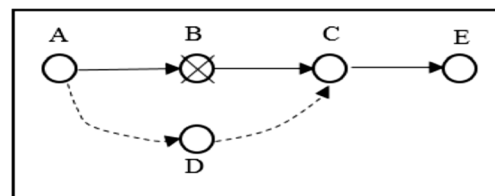


Fig. 3 Illustration of a peer failure

Table 8 Simulation parameters (Scenario 1)

Simulation area size	100 m × 100 m
Number of peers	10–100
Movement patterns	3–5 time intervals
Wireless transmission range	20 m
Number of resources each peer holds	5–10
Number of resource keys used to represent a resource	3–5

we randomly select 60% neighbor peers to flood the requests in our simulations according to the work in [9]. Our simulations are coded in Java SE as did in work [6], and the analyses include the successful search rate, the propagated messages and the search time.

5.1 Simulation settings

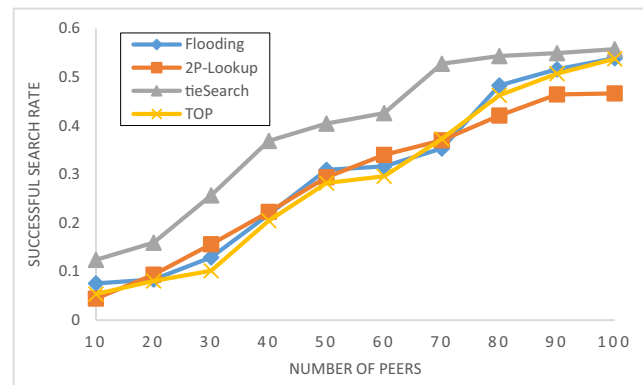
Two scenarios are employed in our simulations. In Scenario 1, the simulation area is set to $100 \times 100 \text{ m}^2$, and we assume that each peer's wireless transmission range is 20 m. There are 10–100 peers in the network, which are randomly distributed in the given simulation area. According to the finding that most peers' movements change regularly on a daily basis [20–22], we have each peer stays online in 3–5 different time intervals in a day. In other words, for each day, each peer may have different neighbor peers in 3–5 different time intervals. Each peer is allocated 5–10 resources, each of which is represented by 3–5 resource keys. Table 8 lists the parameters used in Scenario 1, and Table 9 lists the parameters used in Scenario 2. In a simulation cycle, each peer completes its one day's movements and 100 resource requests are issued by randomly selected peers. To reflect the realistic situation, each peer can move randomly with the probability of 10% without following the regular movements. Each simulation runs 100 cycles and the average value is recorded as the simulation result.

5.2 Successful search rate

In the simulations, we change the number of peers to see the evolutions of the successful search rate. As shown in Figs. 4–5, with the increase of peers, the successful search rates of the four approaches all increase accordingly, in which the

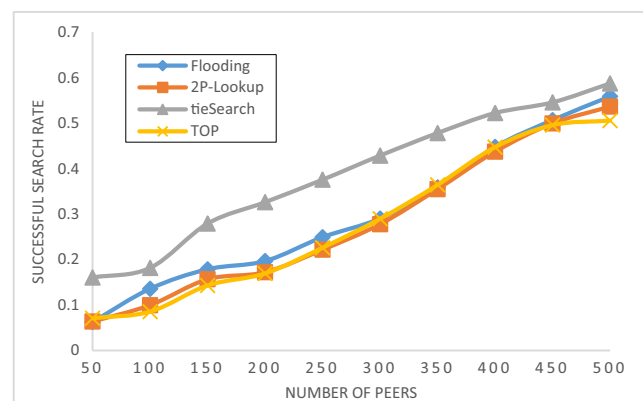
Table 9 Simulation parameters (Scenario 2)

Simulation area size	600 m × 600 m
Number of peers	50–500
Movement patterns	3–6 time intervals
Wireless transmission range	50 m
Number of resources each peer holds	5–10
Number of resource keys used to represent a resource	3–5

**Fig. 4** Evolutions of successful search rates under Scenario 1

tieSearch strategy's successful search rate always outperforms other strategies, regardless of using Scenario 1 or Scenario 2.

In the *tieSearch* strategy, we employ time-aware neighbor peer's availability and time-aware neighbor peer's resource preference to be the pheromones to design our resource search strategy with the ant colony optimization, where the time-aware neighbor peer's availability pheromone can guarantee that the requests are sent or forwarded to the peers being online with high probability in the current time interval and the time-aware neighbor peer's preference pheromone can guarantee that the requests are sent or forwarded to the peers that hold the requested resources with high probability. Therefore, our strategy can achieve high successful search rate in highly dynamic MANETs, as shown in Figs. 4–5. As for the strategy of *2P-Lookup*, since it employs the resource popularity to determine the number of walkers for using the resource popularity-biased random walk approach, its effectiveness cannot be guaranteed due to the peers' strong mobility in MANETs. Similarly, though the *TOP* strategy uses peers' trust to select a neighbor peer to which the request is sent, it does not tell us how to find the peer that owns another peer's trust. The flooding approach floods the requests to the randomly selected 60% neighbor peers without considering whether the selected neighbor peers are being online or not in the current time interval, its successful search rate cannot be guaranteed.

**Fig. 5** Evolutions of successful search rates under Scenario 2

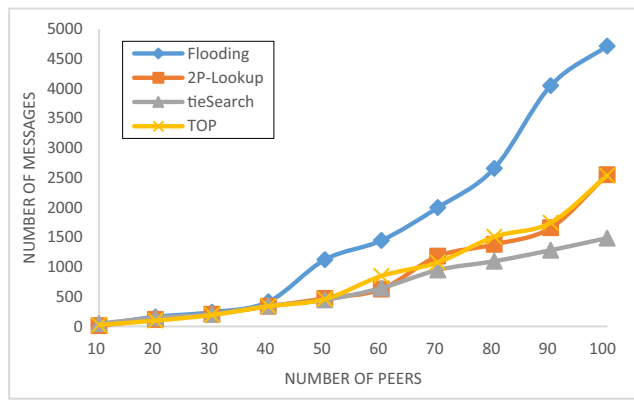


Fig. 6 Evolutions of the propagated messages under Scenario 1

5.3 Propagated messages

This metric is important for MANET environments since the high number of propagated messages means the high consumption of the network bandwidth. We know that a MANET has limited wireless bandwidth, implying that we should take the propagated messages as an important factor to design resource search strategy in MANETs. As shown in Figs. 6-7, when the network has a few number of peers, the numbers of the propagated messages of the four approaches have no significant difference. This is because in such case the number of each peer’s one-hop neighbor peers is fewer, leading to the situation that the previous search experience cannot be fully used to reduce the number of the propagated messages. However, as shown in Figs. 6-7, with the increase of peers in the network, the number of the propagated messages of using the *tieSearch* strategy is lower than those of other three strategies.

Our strategy employs the ant colony model to implement the resource search strategy in MANETs, where we use the time-aware neighbor peer’s availability and the time-aware neighbor peer’s resource preference as the pheromones. The two types of pheromones can guarantee with high probability that a resource request is sent or forwarded to the neighbor peer that is being

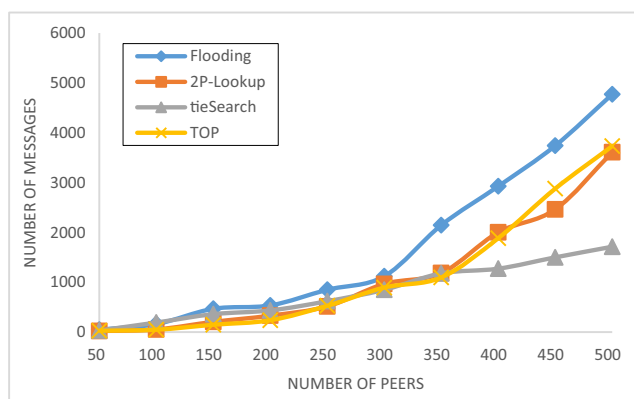


Fig. 7 Evolutions of the propagated messages under Scenario 2

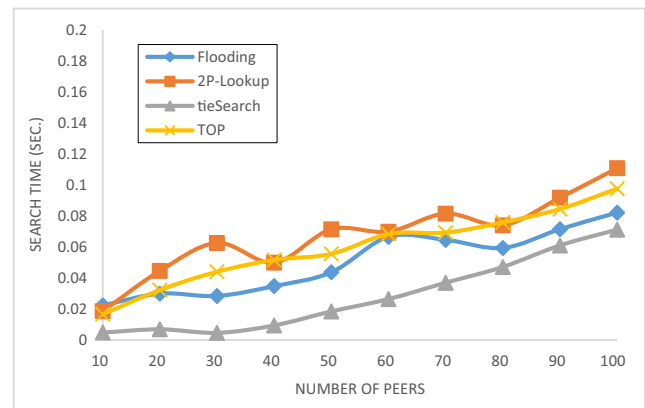


Fig. 8 Evolutions of search time under Scenario 1

online in the current time interval and is also the holder of the requested resource while the request is only sent or forwarded to one neighbor peer. Hence, our strategy propagates lower number of the messages. The *2P-Lookup* approach uses the resource popularity to calculate the number of walkers when it employs the random walk algorithm to send or forward resource requests without taking the time factor into account. Also, *2P-Lookup* needs constructing a P2P overlay based on peers’ physical proximity over MANETs, and a peer’s proximity is detected by broadcasting connection messages. Recall that most peers’ movements change regularly on a daily basis [20–22]. This finding means a peer may have different neighbor peers in different time intervals. In such situation, the *2P-Lookup* approach propagates higher number of messages, as shown in Figs. 6-7. As for the *TOP* strategy, it adopts the following approach to find another peer’s trust. Suppose peer A has previously interacted with peer B. Peer C would ask peer A about peer B’s trustworthiness when it wants to send a request to peer B, as it does not have interaction history with peer B. The question is how peer C knows that peer A has peer B’s trust, making the *TOP* strategy propagate higher number of messages.

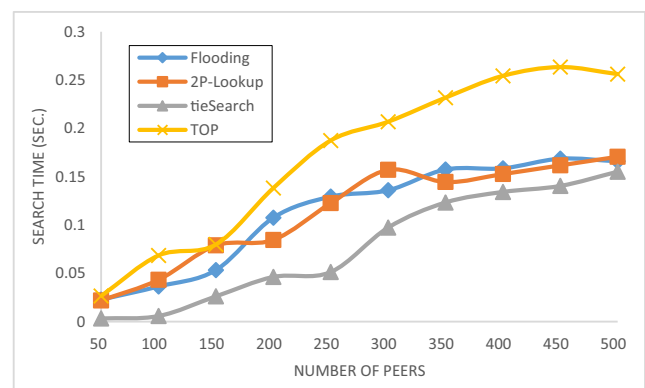


Fig. 9 Evolutions of search time under Scenario 2

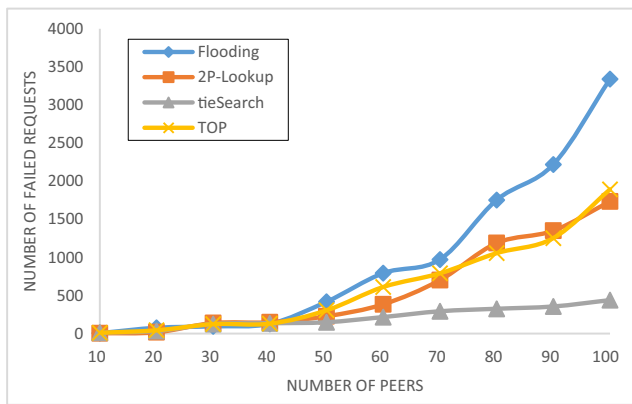


Fig. 10 Evolutions of failed requests under Scenario 1

5.4 Search time

This simulation examines the evolutions of the average search time spent in a successful resource search. The shorter search time implies that the requesters can get their requested resources faster. As shown in Figs. 8-9, the *tieSearch* strategy always needs shorter search time, though the search time of the three strategies has no significant difference. The *tieSearch* strategy sends or forwards an ant bringing the request message to the neighbor peer whose online probability in the current time interval is high and resource preference is highly similar to that the ant is searching for. Therefore, the search time of our strategy is lower.

5.5 Number of failed requests

This metric may impact requesters' experience of the resource search in MANETs. The lower number of failed requests means the requesters can achieve higher resource search success rate. As shown in Figs. 10-11, our strategy has lower number of failed requests. This is because as mentioned above the *tieSearch* strategy sends or forwards requests to the peers whose online probability in the current time interval is high and resource preference is highly similar to that the ant is searching for. Such requests sending or forwarding approach

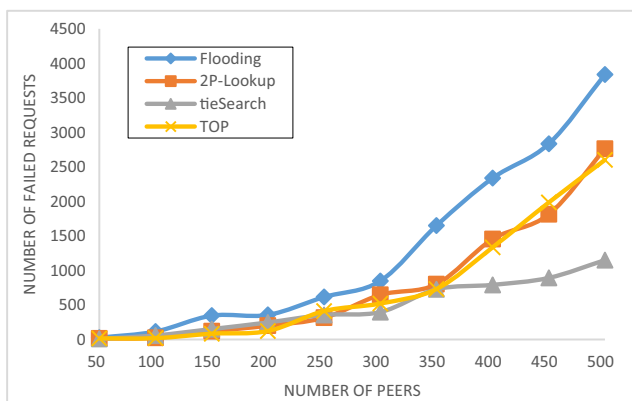


Fig. 11 Evolutions of failed requests under Scenario 2

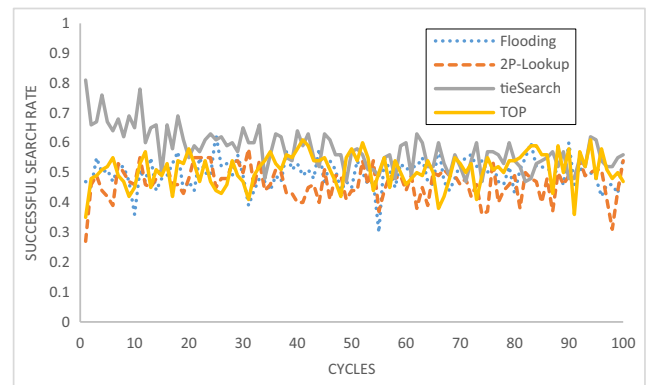


Fig. 12 Successful search rate over simulation cycles under Scenario 1

can lower the number of the failed requests. Though the *2P-Lookup* strategy uses the resource's popularity to determine the number of walkers, since it does not consider the finding that a peer's neighbor peers may be changed in different time intervals, the number of its failed requests is higher than that of *tieSearch*. Similarly, since the *TOP* strategy only uses the trust to select a neighbor peer for forwarding a request without taking the peer's preferences into account, it has higher number of failed requests.

5.6 Evolutions of successful search rate over simulation cycles

This simulation examines the evolutions of the successful search rate over simulation cycles. In the simulations, the number of peers in Scenario 1 and Scenario 2 are fixed at 100 and 500 respectively. As shown in Figs. 12-13, in the initial phase of the simulation, the *tieSearch* strategy has higher successful search rate due to the fact that this phase uses the flooding approach to search resources. From 20th simulation cycle, our strategy's successful search rate enters its steady state. Also, the *tieSearch* strategy always has higher successful search rate than other strategies in any simulation cycle regardless of using Scenario 1 or Scenario 2.

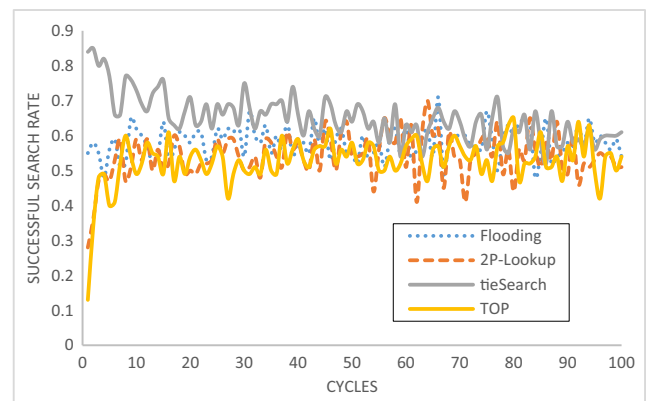


Fig. 13 Successful search rate over simulation cycles under Scenario 2

6 Conclusions

In this paper, we discussed how to improve resource search success rate without increasing network overhead and search time in MANETs. We adopted the ant colony optimization approach to implement our resource search strategy, where we employed the time-aware neighbor peer's availability and the time-aware neighbor peer's resource preference to be the pheromones. The two types of pheromones can guarantee with high probability that a resource request is sent or forwarded to the neighbor peer that is being online in the current time interval and the holder of the requested resource while the request is only sent or forwarded to one neighbor peer. We depicted the initial value calculations of the two types of pheromones, and described the update and attenuation of the pheromones. Then, we detailed our resource search strategy with the ant colony optimization, including the corresponding procedures of the initializing phase, the searching phase and the returning phase an ant foraging process involves. The simulation results showed that our strategy could improve resource search success rate, reduce network overhead and shorten search time. In the future work, we will further focus on the update and attenuation mechanisms of pheromones to improve the performance of our resource search strategy in MANETs.

References

- Ashwin M, Kamalraj S, Azath M (2016) Weighted Clustering Trust Model for Mobile Ad Hoc Networks. *Wirel Pers Commun* 94(4):1–10
- Chen IR, Guo J (2015) Hierarchical trust management of community of interest groups in mobile ad hoc networks. *Ad Hoc Netw* 33:154–167
- Xu L, Wang J, Liu Y et al (2018) Outage Performance for IDF Relaying Mobile Cooperative Networks. *Mobile Networks & Applications* 23(6):1496–1501
- Tarique M, Tepe KE, Adibi S et al (2009) Survey of multipath routing protocols for mobile ad hoc networks. *J Netw Comput Appl* 32(6):1125–1143
- Waluyo AB, Taniar D, Rahayu W et al (2017) Trustworthy data delivery in mobile P2P network. *J Comp Syst Sci* 86:33–48
- Agustinus BW, David T, Wenny R et al (2013) Mobile peer-to-peer data dissemination in wireless ad-hoc networks. *Inf Sci* 230:3–20
- Sasaki Y, Hara T, Nishio S (2014) Top-k query processing for replicated data in mobile peer to peer networks. *J Syst Softw* 92(6):45–58
- Lacuesta R, Penalver L, Fernandez-Sanz L et al (2009) Software Requirements for Ubiquitous Ad Hoc Mobile Networks: An Example of a Bluetooth Application. *International Conference on Software Engineering Advances*. IEEE, 179–184
- Arunachalam A, Sornil O (2015) Issues of Implementing Random Walk and Gossip Based Resource Discovery Protocols in P2P MANETs & Suggestions for Improvement. *Proc Comput Sci* 57: 509–518
- Sarma AD, Molla AR, Pandurangan G (2015) Efficient random walk sampling in distributed networks. *J Parallel Distrib Comput* 77:84–94
- Pu IM, Stamate D, Shen Y (2014) Improving time-efficiency in blocking expanding ring search for mobile ad hoc networks. *J Discrete Algorithms* 24:59–67
- Xu D, Nahrstedt K, Wichadakul D (2001) QoS-Aware Discovery of Wide-Area Distributed Services. *IEEE/ACM International Symposium on CLUSTER Computing and the Grid*, 2001. Proceedings. IEEE, 92–99
- Liang JC, Chen JC, Zhang T (2011) An adaptive low-overhead resource discovery protocol for mobile ad-hoc networks. *Wirel Netw* 17(2):437–452
- Mei H, Zhang Y, Meng X (2014) A path tracking search algorithm based on the credibility of node service ability. *Wireless Communications and NETWORKING Conference*. IEEE, 3385–3389
- Hao L, Xuelong Z (2017) Efficient Resource Search Mechanism in Selfish Mobile Peer-to-Peer Network. *J syst simul* 29(5):1093–1102
- Mondal A, Madria SK, Kitsuregawa M (2006) CLEAR: an efficient context and location-based dynamic replication scheme for mobile-p2p networks. *International Conference on Database and Expert Systems Applications*. Springer-Verlag, 399–408
- Kantero V, Tsoumakos D, Sellis T et al (2009) GrouPeer: Dynamic clustering of P2P databases. *Inf Syst* 34(1):62–86
- Seddiki M, Benchaïba M (2016) 2P-Lookup: Popularity and Proximity based P2P Lookup mechanism over MANETs. *J Netw Comput Appl* 71:181–193
- Zhang H, Jin H, Nie JW et al (2006) Dual-Chord: a More Effective Distribute Hash Table. *J Chinese Comput Syst* 27(8):1450–1454
- Henderson T, Kotz D, Abyzov I (2008) The changing usage of a mature campus-wide wireless network. *Comput Netw* 52(14): 2690–2712
- Orlinski M, Filer N (2013) The rise and fall of spatio-temporal clusters in mobile ad hoc networks. *Ad Hoc Netw* 11:1641–1654
- Ma W, Fang Y, Lin P (2007) Mobility management strategy based on user mobility patterns in wireless networks. *IEEE Trans Veh Technol* 56(1):322–330
- Nayeem MT, Tanvee MM, Hoque R et al (2011) A New Clustering Scheme for Peer-to-Peer File Searching in MANET. *International Journal of Computer Science Issues*
- Atsan E, Özkasap Ö (2013) SCALAR: Scalable data lookup and replication protocol for mobile ad hoc networks. *Comput Netw* 57: 3654–3672
- Pushpalatha M, Ramarao T, Venkataraman R (2014) Applicability of sub graph centrality to improve data accessibility among peers in MANETs. *Peer-to-peer networking and applications* 7:129–146
- Vijayalakshmi P, Francis S, Dinakaran J (2016) A robust energy efficient ant colony optimization routing algorithm for multi-hop ad hoc networks in MANETs. *Wirel Netw* 22(6):2081–2100
- Singh G, Kumar N, Verma A (2014) OANTALG: An Orientation Based Ant Colony Algorithm for Mobile Ad Hoc Networks. *Wirel Pers Commun* 77(3):1859–1884
- Rehman RA, Hieu TD, Bae HM et al (2016) Robust and Efficient Multipath Interest Forwarding for NDN-based MANETs. *2016 9th Ifip Wireless and Mobile Networking Conference*, Colmar, France, Jul 11–13, 187–192
- Hsiao H-C, Su H-W (2012) On optimizing overlay topologies for search in unstructured peer-to-peer networks. *IEEE Trans Parallel Distrib Syst* 23(5):924–935
- Meng X, Wang Y, Gong Y (2015) Perspective of space and time based replica population organizing strategy in unstructured peer-to-peer networks. *J Netw Comput Appl* 49:1–14

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Xianfu Meng received his B.E. and M.E. degrees in Computer Science and Technology from Dalian University of Technology, China in 1983 and 1986, respectively. His current research interests include the peer-to-peer computing and the mobile systems.



Yu Deng is now working toward the master degree in the school of Computer Science and Technology, Dalian University of Technology, China. Her research interests focus on the resource search scheme in mobile ad hoc networks.